# Automatisiertes Auffinden von Präfix- und Suffix-Inklusionsabhängigkeiten in relationalen Datenbankmanagementsystemen

Exposé für eine Diplomarbeit

Jan Hegewald
Betreut von Jana Bauckmann

7. März 2007

## 1 Hintergrund

Das Projekt *Aladin* [LN05] verfolgt das Ziel Datenbanken der Life Sciences weitgehend automatisch zu integrieren. Die Integration erfolgt hierbei weder Schema-orientiert noch mittels einer manuellen Datenanalyse, sondern über die automatische Erkennung von Abhängigkeiten zwischen Objekten. Die integrierten Daten lassen sich dann durch Verfolgen der Beziehungen zwischen den Objekten browsen oder mittels einer Anfragesprache abfragen.

## 2 Aufgabenstellung

Typische Life-Sciences-Datenbanken beschreiben jeweils einen biologischen Sachverhalt, beispielsweise DNA oder Proteine. Zwischen einzelnen Datenbanken bestehen Zusammenhänge. So verweist etwa eine Datenbank über Krankheiten typischerweise von einer Krankheit auf beteiligte Gene in einer Gendatenbank. Diese semantischen Zusammenhänge sind nicht explizit syntaktisch deklariert, sondern müssen zunächst ermittelt werden.

Die verschiedenen Datenquellen können als relationale Datenbanken repräsentiert werden, deren Struktur einem Star Schema ähnelt. Die jeweils beschriebenen Objekte, wie beispielsweise Proteine, werden in der Faktentabelle, hier *Primärrelation* genannt, abgelegt und durch einen Primärschlüssel, die sogenannte *Accession Number*, identifiziert. Zusätzliche Eigenschaften dieser Objekte werden in Dimensionstabellen gespeichert. Im Rahmen der Diplomarbeit sollen Beziehungen zwischen beliebigen Attributen aus einer Datenquelle und den Accession Numbers der Primärrelation einer anderen Datenquelle automatisch identifiziert werden.

Die zu betrachtenden Datenquellen sind in der Regel mehrere Gigabyte groß und bestehen jeweils aus sehr vielen Attributen, die wiederum sehr viele Werte beinhalten. Für die beschriebene Aufgabe werden jedoch nicht einzelne Datenbanken, sondern alle zu integrierenden Datenbanken betrachtet. Um Beziehungen zwischen Attributen zu entdecken, müssen im Allgemeinen alle Paare von

Attributen auf eine Beziehung zueinander geprüft werden. Im vorliegenden Fall ist es voraussichtlich möglich nur die Accession Number der Primärrelation mit allen anderen Attributen aller anderen Datenquellen auf eine Beziehung zueinander zu testen, da bei einer Beziehung von einer Datenquelle zu einer anderen meist die Accession Number der Primärrelation, also das jeweilige biologische Objekt, referenziert wird. Selbst unter diesen Umständen muss jedoch eine sehr hohe Zahl von Tests zwischen zwei Attributen durchgeführt werden, die jeweils alle Tupel einbeziehen müssen, sofern es sich um eine Beziehung handelt. Daher besteht eine besondere Herausforderung der Arbeit darin, die semantischen Zusammenhänge sehr effizient zu finden.

# 3 Vorgehen

## 3.1 Definitionen

Als Beziehungen sollen im konkreten Fall Präfix- und Suffix-Inklusionsabhängigkeiten, (Prefix-Suffix-Inclusion-Dependencies, *PS-INDs*), betrachtet werden. Die im Folgenden eingeführten Begriffe werden in Abbildung 1 an Hand eines Beispiels visualisiert. Eine Inklusionsabhängigkeit zwischen zwei Attributen ist im Allgemeinen so definiert, dass die Menge der Werte des einen Attributes $A$ in der Menge der Werte des anderen Attibutes $B$ enthalten sein muss.

$$A \subseteq B$$

PS-INDs seien nun so definiert, dass eine Menge von Werten eines Attributs in der Menge der Bildwerte einer Funktion $f$, angewandt auf ein anderes Attribut, enthalten ist. Ist die Abbildung $f$ eine Funktion, die eine Zeichenkette um ein Präfix oder um ein Suffix ergänzt, so handelt es sich um eine Präfix- bzw. Suffix-Inklusionsabhängigkeit und damit wahrscheinlich um einen semantischen Zusammenhang der zwei Attribute.

Typischerweise treten PS-INDs zwischen dem Attribut, das die Accession Numbers enthält, und einem Attribut einer Relation in einer anderen Datenbank auf. Dasjenige Attribut, das die Accession Numbers enthält, wird hierbei als *referenziertes Attribut* bezeichnet, das andere als *abhängiges Attribut*. Die Werte der Attribute werden analog bezeichnet.

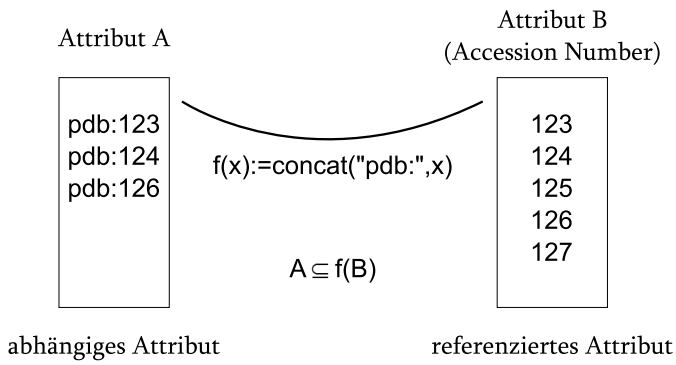


Abbildung 1: Beispiel für eine Präfix-Inklusionsabhängigkeit

## 3.2 Herangehensweise

Eine mögliches Vorgehen zum Finden von PS-INDs besteht darin, zunächst als Abbildung $f$ nur das Anhängen eines Suffixes $s$ zu betrachten.

$$f(x) := concat(x, s)$$

Um zu testen, ob eine PS-IND vorliegt, reicht es dabei aus, zu prüfen, ob die Menge aller Werte des Attributes $A$ eine Teilmenge der Werte des Attributes $B$ verkettet mit dem Suffix $s$ bildet.

$$A \subseteq concat(B, s)$$

Im Detail wird dies im folgenden Abschnitt 3.3 beschrieben.

Als nächstes soll der Fall betrachtet werden, in dem man PS-INDs finden möchte, bei denen die Abbildung $f$ ein Präfix $p$ vor den referenzierten Wert stellt.

$$f(x) := concat(p, x)$$

In diesem Fall ist zu prüfen, ob die Inklusionsbeziehung

$$A \subseteq concat(p, B)$$

gilt. Definiert man die Operation $(\circ)^{-1}$ als Umkehrung eines Strings, so ist die obige Beziehung äquivalent zu:

$$A^{-1} \subseteq concat(B^{-1}, p^{-1})$$

Somit kann man das Präfix-Problem auf das Suffix-Problem zurückführen und jeder Algorithmus, der das Suffixproblem löst, löst auch das Präfixproblem.

Die geforderte „echte“ Inklusionsbeziehung tritt in der Praxis möglicherweise selten auf, da Daten oft verschmutzt sind. Um dennoch PS-INDs zu finden, müsste der Algorithmus so modifiziert werden, dass er einen festgelegten Anteil von „nicht-abhängigen Werten“ toleriert und dennoch eine PS-IND zwischen zwei Attributen erkennt.

## 3.3 Teilaufgaben

Eine Teilaufgabe beim Finden von PS-INDs ist das Erkennen möglicher Präfixe oder Suffixe. Hierbei ist zunächst zu definieren, welche Typen in Frage kommen. Es ist denkbar Affixe variabler Länge zu erkennen, wenn die Länge der Accession Number konstant ist. Dies ist auch ein praktikabler Ansatz, wenn der referenzierte Wert aus anderen Zeichen besteht als das Affix, beispielsweise aus numerischen und alphabetischen Zeichen. Alternativ könnte man Affixe fester Länge suchen. Dieser Fall ist einfacher zu behandeln, wenn sich die Länge der Affixe zuverlässig ermitteln lässt. Für die Erkennung kommen verschiedene Ansätze in Betracht: Pattern-Mining-Algorithmen, Verfahren, die auf Datenbankstatistiken beruhen, oder selbst zu entwickelnde Ansätze.

Wurde ein wahrscheinliches Affix ermittelt, muss ein Algorithmus gefunden werden, der entsprechend den oben dargelegten Überlegungen die Inklusionsbeziehung zwischen den Werten der zwei Attribute überprüft.

Hierbei wird es bei Affixen fester Länge wahrscheinlich zielführender sein, beim Vergleich nicht die referenzierten Werte um das Affix zu ergänzen, sondern

dieses vom abhängigen Wert zu entfernen und anschließend die Inklusionsbeziehung zu prüfen. Für den Fall variabler Affixe müssen andere Wege zum Testen der Inklusionsbeziehung gefunden werden. Sofern möglich, soll der bestehende Algorithmus *SPIDER* [BLNT07] für das Finden der PS-INDs angepasst und verwendet werden.

Da die zu untersuchenden Datenquellen mehrere GBytes groß sind, muss ein Algorithmus zur Lösung dieses Problems sehr effizient sein. Daher sind Komplexitätsabschätzungen unabdinglich.

Weiterhin sind Heuristiken zu entwickeln, die syntaktisch scheinbare, aber semantisch unwahrscheinliche Abhängigkeiten erkennen und somit ausschließen können.

# 4 Erweiterungsmöglichkeiten

Bisher wurde angenommen, dass die Abbildung *entweder* Präfixe *oder* Suffixe an den referenzierten Wert anhängt. Gibt man diese Annahme auf, so funktioniert das oben vorgeschlagene Verfahren eventuell noch, wenn man die Suffixe und Präfixe identifizieren und abschneiden kann. Ist dies nicht möglich, so wäre ein Zeichenkettentest auf das Enthaltensein der referenzierten Werte in den abhängigen Werten anzuwenden. Die Laufzeit des Algorithmus dürfte in diesem Fall jedoch wahrscheinlich (viel) schlechter werden, da für jeden Wert des potenziell referenzierten Attributs getestet werden muss, ob irgendein Wert des potenziell abhängigen Attributs existiert, der den referenzierten Wert als Teilstring enthält.

Außerdem kann man die Annahme aufgeben, dass ein Attribut *genau eine* semantische Beziehung zu einem anderen Attribut in einer anderen Relation aufweist. Statt dessen könnten die abhängigen Werte Accession Numbers verschiedener Datenquellen referenzieren. Hier müssten dann Gruppen von abhängigen Werten gefunden werden. Eine Gruppe ist eine Teilmenge der Werte eines Attributs, die für sich betrachtet eine PS-IND zu einer Accession Number aufweist. Kann diese minimale Anzahl/Anteil gefunden werden, so deutet dies auf eine Beziehung zwischen den Tupeln der beteiligten Relationen hin. Dieser Fall ist ungleich schwieriger. Mindestens zwei Ansätze sind möglich. Zunächst könnte man das Problem eventuell auf das verschmutzter Daten reduzieren. Hierbei müsste man den zulässigen Anteil der „nicht-abhängigen Werte“ hoch ansetzen und das abhängige Attribut auf eine PS-IND mit weiteren Attributen testen, auch wenn bereits semantische Beziehungen zu Primärrelationen gefunden wurden. Fraglich ist dann jedoch, ob es noch zuverlässig möglich sein wird Affixe in der Gesamtmenge der Werte des Attributs zu erkennen. Alternativ könnte man zunächst versuchen die einzelnen Gruppen zu identifizieren und dann jede einzelne auf eine PS-IND mit anderen Attributen zu testen. Diese Partitionierung des abhängigen Attributs könnte wahrscheinlich nur über Pattern-Mining-Algorithmen erreicht werden. Da Pattern-Mining-Algorithmen jedoch wahrscheinlich eine recht hohe Komplexität haben, ist es fraglich, ob dieser reine Vorverarbeitungsschritt in für die Datenmengen hinreichender Performanz umgesetzt werden kann.

# Literatur

[BLNT07] Jana Bauckmann, Ulf Leser, Felix Naumann, and Véronique Tietz. Efficiently detecting inclusion dependencies. In *Proceedings of the International Conference on Data Engineering (ICDE 2007)*, 2007.

[LN05] Ulf Leser and Felix Naumann. (Almost) Hands-off information integration for the life sciences. In *Proceedings of the Conference on Innovative Database Research (CIDR 2005)*, 2005.